International Journal of Pure and Applied Mathematics
Volume 119 No. 12 2018, 13287-13295
ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

SENTIMENT ANALYSIS USING STRING TOKEN CLASSIFICATION ALGORITHM

Subramaniyaswamy V*, Harshaa S, Padma Janani M, Prabhalammbeka B S
School of Computing, SASTRA Deemed University, Thanjavur, India
*Corresponding Author

Abstract: Sentiment analysis is a type of data mining which involves computation of opinions, sentiments and to determine if an information or a piece of text conveys positive, negative or neutral opinion. Public opinion regarding various aspects can be found using sentiment analysis. Clustering and classification are the key techniques in sentiment analysis. Consensus clustering is better than existing clustering algorithms as it provides a stable and efficient final result. However, it has its own drawbacks. Instead of performing consensus clustering and selecting classifiers from the consolidated result, we try to develop a new classification algorithm in our work

Index Terms: Consensus Clustering, Sentiment Analysis, String Token Classifier, Text Classification

1. Introduction

In sentiment analysis, Consensus clustering has been proposed as the technique for performing clustering in our base paper which deals with overcoming the instability of the existing clustering algorithms by integrating the base algorithms in order to form a more robust clustering algorithm [16-26]. By doing so, a more stable final clustering result will thus be obtained thus eliminating the drawbacks present and also better than the conventional ensemble pruning and clustering methods present already [27-35].

However, this mechanism can only be applied where the data sets do not contain string values.Taking Weka into consideration, we will find that only hierarchical clustering is the only methodology(type) applicable for data set with such attributes and the rest of the algorithms are not compatible for processing it and throws error when we try to process it [36-40]. Hence, consensus clustering does not fare well with string values. So, consensus clustering is not suitable for the analysis of dataset which contain strings. (cannot be implemented in this case.) Instead of performing consensus clustering and then later on picking a candidate or couple of candidate classifiers from it based on the predictive characteristics, we have developed a new classification algorithm which does not require clustering prior to it.

The objective is to obtain better results with higher efficiency when compared with the existing classification algorithms which are already exist in Weka software. Implementing consensus clustering requires the presence of more than one clustering algorithm as discussed above and we have already mentioned that it is not possible to do the clustering in Weka as it is supported by only one algorithm that works with string values which involves balanced and unbalanced value types. Clustering is used to group objects to different groups and also to find the structure in collection of data that are not labelled already (i.e) collection of similar objects which are not similar to the objects of other clusters.

Data clustering algorithms can either be hierarchical or partition. Partition algorithms can determine all clusters at a time whereas Hierarchical algorithms can find successive clusters using previous clusters and hierarchical algorithms can either be agglomerative (bottom-up) or divisive (top-down). The classification algorithm which has been developed is known as string token classification algorithm.

2. Related Works

Classification is a process of data mining that categorizes a set of items according to its relativity. The efficiency of a classifier process lies on how accurately it categorizes the item. A classification algorithm finds the relationships between the worth of the predictors and the values of the goal. Five different classification algorithms j48, random tree, Bayesian network, rep tree and logistic model are studied and

analyzed in this paper and their performance is evaluated by precision, recall and F measure for liver disorder dataset. The result of this analysis suggests that random tree provides high accuracy than the other algorithms [9].

Data mining is a field of research where vast amounts of data are analyzed and understood to identify the hidden information pattern in it. Jumping Emerging Pattern(JEP) is a new knowledge model and there is a classification algorithm based on it. In this paper a special type of JEP, known as SJEP(Most Significant Jumping Emerging Pattern) is identified. This algorithm has the ability to distinguish between JEP strong, while using only the most effective jumping emerging patterns (SJEP) as a basis for classification algorithm enhances endurance noise, reducing the complexity of the algorithm [15]. SVM is one of the best classifier where maximum accuracy and minimum root mean square error can be obtained. They compare SVM kernel types and identify radial basis kernel as the best of SVM [1]. Decision tree is one of the efficient classification methods in data mining and ID3 is the most commonly used algorithm in it. Featured attributes of a dataset are initially segregated into groups and then selection measure is applied on them. This step is repeated until we get efficient and accurate classification algorithm [10].

As most of the information in social media are text oriented, text mining has a higher potential value in research involving data from social media. The first step of text mining is text classification where a document will be classified in terms of some categories. In this paper, some of the existing classifiers are compared based on certain criteria to give an overview to text classifiers [5]. Extraction of data from vast amounts of databases is known as Data Mining. It is a new technology which helps in focusing on most important information. This paper analyses some of the classification algorithms such as Linear Regression, Multi-Layer Perceptron, CART, J48, C4.5, ID3, Random forest and KNN [12].

To analyses and extract useful information from a large volume of data, data mining techniques are employed. Educational institutions also use such techniques to improve the performance of students. this study presents an analysis of every semester results of UG degree students using data mining technique. it compares the result of classification algorithms. It identifies the suitable algorithm for predicting the performance of the students among the selected algorithms. The analysis work is done by considering various types of algorithm like decision tree algorithm, rule-based algorithm, Bayesian algorithm and function-based algorithms. This generic novel approach can be extended to other disciplines as well [11]. One of the rapid growing research area is Collaborative filtering. A variety of collaborative filtering techniques are compared to identify apt algorithm for a condition. Here both classic and recent methods of collaborative filtering are compared to achieve better results [7].

3. Materials and Methods

The primary step involves the loading of the data set and then it works in such a manner that all the balanced and the unbalanced data set values with reviews (for mobiles) contain positive, negative and neutral reviews alike. It takes each sentence (all the words) present in the data set in the review section and then splits each of the words present in the form of tokens. The occurrence of these tokens in the whole data set are counted in such a way that the count of the occurrence of each token in a positive and negative feedback (in balanced dataset) or positive, negative and neutral (in unbalanced dataset) are collected separately. Finally, the word frequency of the tokens is calculated.

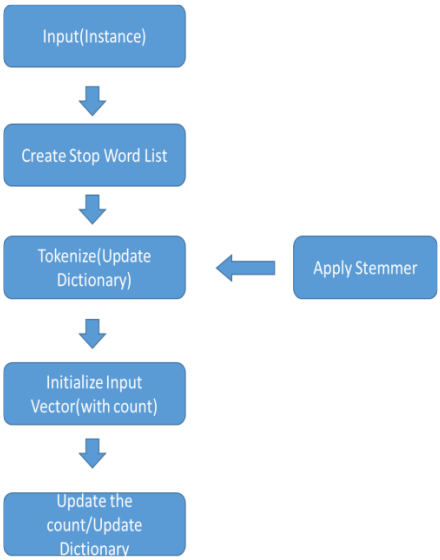

Figure 1. The general structure of the proposed String Token Classification Algorithm.

3.1 Theoretical Foundations

Tokenizer and stemmer are involved in the process. Tokenizer helps in the process of converting each sentence into sequence of tokens. Tokenization depends on simple heuristics for the purpose of separating tokens. Basically, tokens are separated by whitespace, punctuations marks and line breaks. Whitespace and punctuations may or may not be included depending on our needs. Tokens can be made up of all alpha character, alpha numeric characters or numerals. Later, the tokens count is calculated by LinkedHashMap method. Stemming is a method used for grouping words with similar basic meaning together. Stemming is an automated process of reducing derived words to their root form.

The LinkedHashMap is a map interface which is used to predict the iteration order. A LinkedHashMap contains values based on keys (K (key, V(Value)). It uses the hash table and linked list for the implementation of map interface. The one big advantage in using LinkedHashMap is it contains only the unique elements. Similarly, Serialization is an important technique used in this classification. Serialization is a process which involves writing of the object's state into the sequence of the byte stream. It is used for marshaling, to travel object's state inside the classifier.

With the above techniques mentioned above, the missing values, numeric attributes, date attributes, values which will not affect the outcome (stop words) are ignored while classifying and the lower case, upper case letters, word frequencies and also the minimum number of instances are considered and all the values are normalized. The undesirable attributes are disabled when encountered in the classifying process by including exception that are both user defined as well as pre-defined.

Input: m_data
m_data, Data Instance
1: C ←0
2: Ok ← false
3: Count←0
4: m_probOfWord ←0
5: word ← null
6: buildClassifier(Instance Data)
7: m_probOfWord←HashMap<Integer, LinkedHashMap<String, Count>>
8: updateClassifier(data Instance, Boolean updateDictionary)
9: distributionForInstance(Instance instance)
10: LinkedHashMap<String, Count>
11: If(Normalize)
12: while (Map.Entry<String, Count> feature)
13: Word←feature.getkey()
14: Count← feature.getvalue()
15: End while
16: Loop 0 to m_data.numClass
17: m_probOfWord!= null
18: ok← true
19: return classifiedValues

Figure 2. The general structure of String Token Classification Algorithm.

1: updateClassifier(data Instance, Boolean updateDictionary)
2: if (!instance.classIsMissing())
3: tokenizeInstance

Figure 3. Update Dictionary

```
tokenizeInstance(Instance,Boolean
updateDictionary)

1:  t, tokens list
2:  d, no.of occurrence(token)
3:  d←0
4:  t←null
5:  Loop m_data.count!=0
6:  t←tokens
7:  d←d+1
8:  setStemmer(Stemmer value)
9:  End loop
```

Figure 4. Tokenize Instance

4. Experimental analysis

For the purpose of analyzing the performance of the proposed String Token Classification Algorithm on sentiment classification, this section gives the dataset used in analysis, the experimental step and results.

4.1 Data Set

Our analysis was conducted using both balanced and unbalanced benchmarks. The balanced dataset is movie review taken from github.com(movie-pang02.zip). It is the review of the Pang and Lee movie. For the unbalanced dataset mobile review is taken from the same github.com(twitter-sanders-apple3.zip).It is the dataset based on review about the iPhone 3 of Apple company.

4.2 Dataset Description

The movie reviews (balanced) have two categories: Positive (reviews that express a favourable sentiment or positive) and Negative (reviews that express a unfavourable sentiment or negative). With respect to our analysis, the reviews contain either positive reviews or negative reviews and the are no neutral reviews. The mobile reviews(unbalanced) has all the three positive, negative and neutral reviews for our analysis. Both the movie and mobile reviews are in the csv format and the reviews have been collected form Twitter tweets.

4.3 Experimental Result

The obtained experimental results are described and tabulated.

4.3.1 Balanced Dataset

The output of the experiment lists the values of the correctly classified instances, incorrectly classified instances, mean absolute error, root mean squared error, relative absolute error and root relative squared error for various classifiers using balanced dataset as input. As it can be observed from Table 1, the highest classified results are obtained from String Token Classification in comparison with the existing classifiers.

Table 1. Descriptive comparison with existing classification algorithms (Stratified cross-validation Summary)

Classifier	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
String Token Classifier (Bayes)	**53.5%**	**46.5%**	**0.4614**	**0.5472**	**92.2897%**	**109.43%**
SGD Text (Functions)	52%	48%	0.48	0.6928	96%	138.56%
Multi Scheme (Meta)	50%	50%	0.5	0.5	100%	100%
Input Mapped Classifier (Misc)	50%	50%	0.5	0.5	100%	100%
K FOLD (Rules)	50%	50%	0.5	0.5	100%	100%

The stratified cross validation results of String Token Classifier and other existing classifiers using balanced datasets as input are represented in Table 2.

Table 2. Detailed Accuracy by Class

Classifier	Class	TP Rate	FP Rate	Precision	Recall	F Measure
String Token Classifier	**Positive**	**0.430**	**0.360**	**0.544**	**0.430**	**0.480**

(Bayes)	**Negative**	**0.640**	**0.570**	**0.529**	**0.640**	**0.579**
	Weighted Avg.	**0.535**	**0.465**	**0.537**	**0.535**	**0.530**
SGD Text (Functions)	Positive	0.490	0.450	0.521	0.490	0.505
	Negative	0.550	0.510	0.519	0.550	0.534
	Weighted Avg.	0.520	0.480	0.520	0.520	0.520
Multi Scheme (Meta)	Positive	1.000	1.000	0.500	1.000	0.667
	Negative	0.000	0.000	0.000	0.000	0.000
	Weighted Avg.	0.500	0.500	0.250	0.500	0.33
Input Mapped Classifier (Misc)	Positive	1.000	1.000	0.500	1.000	0.667
	Negative	0.000	0.000	0.000	0.000	0.000
	Weighted Avg.	0.500	0.500	0.250	0.500	0.33
K FOLD (Rules)	Positive	1.000	1.000	0.500	1.000	0.667
	Negative	0.000	0.000	0.000	0.000	0.000
	Weighted Avg.	0.500	0.500	0.250	0.500	0.33

4.3.2 Unbalanced Dataset

Table 3: Descriptive comparison with existing classification algorithms (Stratified cross-validation Summary)

Classifier	Correctly Classified Instances	Incorrectly Classified Instances	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
String Token Classifier (Bayes)	**67.915 %**	**32.085 %**	**0.2298**	**0.4064**	**56.945 %**	**90.48 %**
Multi Scheme (Meta)	51.5182 %	48.4818 %	0.4035	0.4491	100%	100%
Input Mapped Classifier (Misc)	51.5182 %	48.4818 %	0.4035	0.4491	100%	100%
K FOLD (Rules)	51.5182 %	48.4818 %	0.4035	0.4491	100%	100%

Table 4. Detailed Accuracy by Class

Classifier	Class	TP Rate	FP Rate	Precision	Recall	F Measure
String Token Classifier (Bayes)	**Positive**	**0.160**	**0.004**	**0.897**	**0.160**	**0.271**
	Negative	**0.810**	**0.246**	**0.608**	**0.810**	**0.695**
	Neutral	**0.764**	**0.311**	**0.723**	**0.764**	**0.743**
	Weighted Avg.	**0.679**	**0.239**	**0.715**	**0.679**	**0.650**
Multi Scheme (Meta)	Positive	0.000	0.000	0.000	0.000	0.000
	Negative	0.000	0.000	0.000	0.000	0.000
	Neutral	1.000	1.000	0.515	1.000	0.680
	Weighted Avg.	0.515	0.515	0.265	0.515	0.350
Input Mapped Classifier (Misc)	Positive	0.000	0.000	0.000	0.000	0.000
	Negative	0.000	0.000	0.000	0.000	0.000
	Neutral	1.000	1.000	0.515	1.000	0.680
	Weighted Avg.	0.515	0.515	0.265	0.515	0.350
K FOLD (Rules)	Positive	0.000	0.000	0.000	0.000	0.000
	Negative	0.000	0.000	0.000	0.000	0.000
	Neutral	1.000	1.000	0.515	1.000	0.680
	Weighted Avg.	0.515	0.515	0.265	0.515	0.350

5. Discussion

This section is aimed at describing the performance of different classifiers in comparison with the proposed string token classification algorithm. The proposed algorithm is string data efficient based on classification process. As mentioned above, the existing classification algorithms generally use stemmer, tokenizer, normalizer either separately or any two combined to achieve better results. Here this proposed algorithm uses all those three elements in order to obtain more efficient classified instances.

In this paper, string data (mobile review) is taken as input, since the algorithm operates only on string data. The pre-processed input is classified using the existing enabled classifiers and the obtained correctly classified instances are noted. Similarly, the pre-processed input is classified using the proposed classification algorithm and the obtained correctly classified instances are noted. Both the noted results are analyzed and compared. The results of the comparison evidently prove that the proposed algorithm has higher correctly classified instances than the existing classifiers by showing the numerical outputs

6. Conclusion

Our work describes text sentiment-based classification scheme by using String Token

Classifier algorithm. It overcomes the inability of other classification algorithms which are unable to process string-based data set values and work well only with numerical values. Text present in the data set in the review section is evaluated after instance is created and we form stop words list. The use of stemmer, tokenizer and normalization is aimed at producing classified instances with better efficiency compared to the existing classifiers by taking into account both balanced and unbalanced data sets. The results indicate that the proposed scheme can yield better promising results compared to the conventional methods. Our work can be extended by incorporating collaborative filtering methods for the purpose of recommending products to the users based on the reviews obtained from the data set.

7. References

[1]. Agarwal, S., Pandey, G. N., & Tiwari, M. D. (2012). Data mining in education: data classification and decision tree approach. International Journal of e-Education, e-Business, e-Management and e-Learning, 2(2),140.

[2]. Chen, L., Wang, W., Nagarajan, M., Wang, S., & Sheth, A. P. (2012). Extracting diverse sentiment expressions with target-dependent polarity from Twitter. In Proceedings of the sixth international AAAI conference on weblogs and social media (pp. 50–57).

[3].Elghazel, H. , Aussem, A. , Gharroudi, O. , & Saadaoui, W. (2016). Ensemble multi-label text categorization based on rotation forest and latent semantic index- ing. Expert Systems with Applications, 57, 1–11.

[4].Fusilier, D. H. , Montes-y-Gomez, M. , Rosso, P. , & Cabrera, R. G. (2015). Detecting positive and negative deceptive opinions using PU-learning. Information Processing and Management, 51, 433–443.

[5]. Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications, 3(2),85.

[6]. Khatri, M. (2012). A survey of naïve bayesian algorithms for similarity in recommendation systems. International Journal of Advanced Research in Computer Science and Software Engineering, 2(5).

[7]. Lee, J., Sun, M., & Lebanon, G. (2012). A comparative study of collaborative filtering algorithms. arXiv preprint arXiv:1205.3193.

[8].Mendialdua, I. , Arruti, A. , Jauregi, E. , Lazkano, E. , & Sierra, B. (2015). Classifier subset selection to construct multi-classifiers by means of estimation of distribution algorithms. Neurocomputing, 157, 46–60.

[9]. Parimala, C., & Porkodi, R. (2018). Classification Algorithms in Data Mining: A Survey.

[10]. Parashar, H. J., Vijendra, S., & Vasudeva, N. (2012). An efficient classification approach for data mining. International Journal of Machine Learning and Computing, 2(4), 446.

[11]. Patel, B., & Gondaliya, C. (2017). Student Performance Analysis Using Data Mining Technique.

[12]. Ponmani, S., & Vidhu Priya, P. (2017). Classification Algorithms in Data Mining–A Survey. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume, 6.

[13]. Ting, S. L., Ip, W. H., & Tsang, A. H. (2011). Is Naive Bayes a good classifier for document classification. International Journal of Software Engineering and Its Applications, 5(3), 37-46.

[14]. WEKA 3: (2016). Data mining software in Java (n.d.) Retrieved May 25 from http://www.cs.waikato.ac.nz/ml/weka/.

[15]. Zhao, L., Chen, D. F., Xu, S. J., & Lu, J. (2015). The Research of Data Mining Classification Algorithm that Based on SJEP. International Journal of Database Theory and Application, 8(2), 223-234.

[16]. Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., Gao, X. Z., & Indragandhi, V. (2017). A hybrid quantum-induced swarm

intelligence clustering for the urban trip recommendation in smart city. Future Generation Computer Systems, 83, 653-673.

[17]. Subramaniyaswamy, V., & Logesh, R. (2017). Adaptive KNN based Recommender System through Mining of User Preferences. Wireless Personal Communications, 97(2), 2229-2247.

[18]. Logesh, R., & Subramaniyaswamy, V. (2017). A Reliable Point of Interest Recommendation based on Trust Relevancy between Users. Wireless Personal Communications, 97(2), 2751-2780.

[19]. Logesh, R., & Subramaniyaswamy, V. (2017). Learning Recency and Inferring Associations in Location Based Social Network for Emotion Induced Point-of-Interest Recommendation. Journal of Information Science & Engineering, 33(6), 1629–1647.

[20]. Subramaniyaswamy, V., Logesh, R., Abejith, M., Umasankar, S., & Umamakeswari, A. (2017). Sentiment Analysis of Tweets for Estimating Criticality and Security of Events. Journal of Organizational and End User Computing (JOEUC), 29(4), 51-71.

[21]. Indragandhi, V., Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., Siarry, P., & Uden, L. (2018). Multi-objective optimization and energy management in renewable based AC/DC microgrid. Computers & Electrical Engineering.

[22]. Subramaniyaswamy, V., Manogaran, G., Logesh, R., Vijayakumar, V., Chilamkurti, N., Malathi, D., & Senthilselvan, N. (2018). An ontology-driven personalized food recommendation in IoT-based healthcare system. The Journal of Supercomputing, 1-33.

[23]. Arunkumar, S., Subramaniyaswamy, V., & Logesh, R. (2018). Hybrid Transform based Adaptive Steganography Scheme using Support Vector Machine for Cloud Storage. Cluster Computing.

[24]. Indragandhi, V., Subramaniyaswamy, V., & Logesh, R. (2017). Resources, configurations, and soft computing techniques for power management and control of PV/wind hybrid system. Renewable and Sustainable Energy Reviews, 69, 129-143.

[25]. Ravi, L., & Vairavasundaram, S. (2016). A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. Computational intelligence and neuroscience, 2016, Article ID: 1291358.

[26]. Logesh, R., Subramaniyaswamy, V., Malathi, D., Senthilselvan, N., Sasikumar, A., & Saravanan, P. (2017). Dynamic particle swarm optimization for personalized recommender system based on electroencephalography feedback. Biomedical Research, 28(13), 5646-5650.

[27]. Arunkumar, S., Subramaniyaswamy, V., Karthikeyan, B., Saravanan, P., & Logesh, R. (2018). Meta-data based secret image sharing application for different sized biomedical images. Biomedical Research,29.

[28]. Vairavasundaram, S., Varadharajan, V., Vairavasundaram, I., & Ravi, L. (2015). Data mining-based tag recommendation system: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(3), 87-112.

[29]. Logesh, R., Subramaniyaswamy, V., & Vijayakumar, V. (2018). A personalised travel recommender system utilising social network profile and accurate GPS data. Electronic Government, an International Journal, 14(1), 90-113.

[30]. Vijayakumar, V., Subramaniyaswamy, V., Logesh, R., & Sivapathi, A. (2018). Effective Knowledge Based Recommeder System for Tailored Multiple Point of Interest Recommendation. International Journal of Web Portals.

[31]. Subramaniyaswamy, V., Logesh, R., & Indragandhi, V. (2018). Intelligent sports commentary recommendation system for individual cricket players. International Journal of Advanced Intelligence Paradigms, 10(1-2), 103-117.

[32]. Indragandhi, V., Subramaniyaswamy, V., & Logesh, R. (2017). Topological review and analysis of DC-DC boost converters. Journal of

Engineering Science and Technology, 12 (6), 1541–1567.

[33]. Saravanan, P., Arunkumar, S., Subramaniyaswamy, V., & Logesh, R. (2017). Enhanced web caching using bloom filter for local area networks. International Journal of Mechanical Engineering and Technology, 8(8), 211-217.

[34]. Arunkumar, S., Subramaniyaswamy, V., Devika, R., & Logesh, R. (2017). Generating visually meaningful encrypted image using image splitting technique. International Journal of Mechanical Engineering and Technology, 8(8), 361–368.

[35]. Subramaniyaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A., & Vijayakumar, V. (2017). A personalised movie recommendation system based on collaborative filtering. International Journal of High Performance Computing and Networking, 10(1-2), 54-63.

[36]. Senthilselvan, N., Udaya Sree, N., Medini, T., Subhakari Mounika, G., Subramaniyaswamy, V., Sivaramakrishnan, N., & Logesh, R. (2017). Keyword-aware recommender system based on user demographic attributes. International Journal of Mechanical Engineering and Technology, 8(8), 1466-1476.

[37]. Subramaniyaswamy, V., Logesh, R., Vijayakumar, V., & Indragandhi, V. (2015). Automated Message Filtering System in Online Social Network. Procedia Computer Science, 50, 466-475.

[38]. Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., & Indragandhi, V. (2015). Unstructured data analysis on big data using map reduce. Procedia Computer Science, 50, 456-465.

[39]. Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., & Indragandhi, V. (2015). Intelligent travel recommendation system by mining attributes from community contributed photos. Procedia Computer Science, 50, 447-455.

[40]. Vairavasundaram, S., & Logesh, R. (2017). Applying Semantic Relations for Automatic Topic Ontology Construction. Developments and Trends in Intelligent Technologies and Smart Systems, 48.